Programming Environment 3

3.1 OpenGL Programming Environment

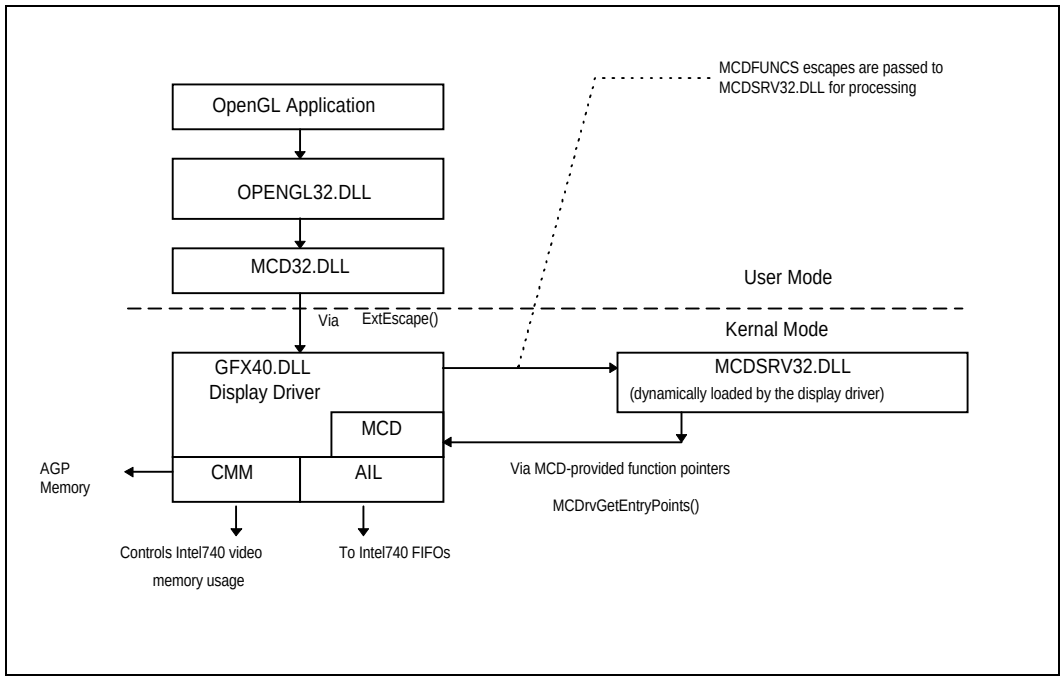
OpenGL is an application programming interface (API) which is used by a software application to interface with the graphics hardware. OpenGL consists of approximately 120 different commands which are used to specify graphical objects and the operations applied to the objects which are required by 3D applications. OpenGL is a streamlined, hardware-independent interface designed to make applications portable from one hardware platform to another. For more information on the OpenGL function commands, see the OpenGL Specification document which can be obtained from the SGI web site at http://www.sgi.com. Also see Section 4.3, “OpenGL Programming Implementation” on page 4-27, for the Intel740™ graphics accelerator-specific OpenGL performance information.

3.2 OpenGL Drivers

3.2.1 MCD

In the MCD, or Microsoft* Mini Client Driver OpenGL implementation for WindowsNT, the operations are split between the Microsoft* portion of the driver (performing such computations as geometry and lighting, etc.) and the Intel740™ graphics accelerator device-dependent portion of the driver. The MCD architecture is described in Figure 3-1.

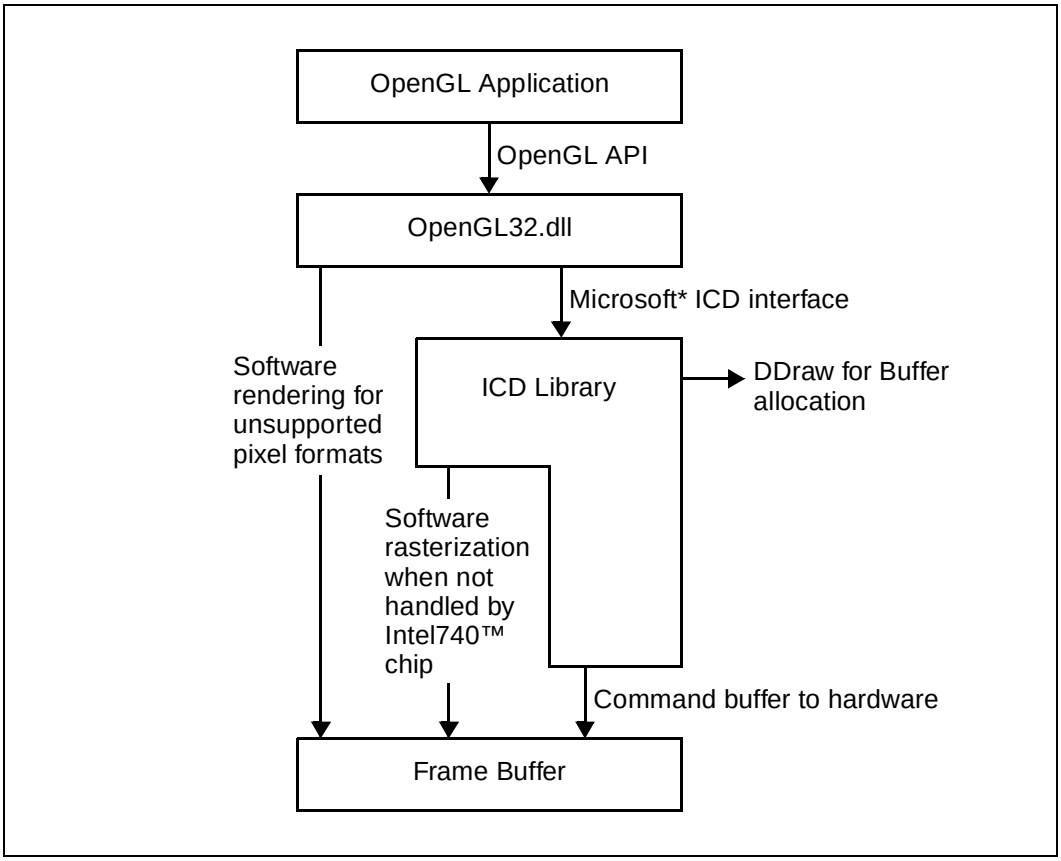
Figure 3-1. MCD Architecture



3.2.2 ICD

The ICD, or Independent Client Driver OpenGL implementation for Windows9x, implements all of the OpenGL function calls including lighting and geometry through a combination of Intel740™ drivers and the Intel740™ chip. The architecture of the ICD is shown in Figure 3-2.

Figure 3-2. ICD Architecture



3.2.2.1 Buffer Allocation

The ICD uses DirectDraw to allocate the back buffer and depth (Z) buffer. Additionally, for full-screen applications, the ICD automatically obtains “exclusive mode” access to the buffers to optimize flip operations (buffer swaps). If enough local video memory is available, the ICD automatically uses triple buffering to prevent stalls caused by buffer swaps. If there is insufficient local video memory to allocate the back buffer(s) or depth buffer, the ICD will resort to software rendering.

3.2.3 Geometry Operations

The geometry engine within the OpenGL drivers for the Intel740 graphics accelerator support the following computations:

- Transformations
- Clipping
- User-defined clipping planes
- Culling
- Lighting
  - — ColorMaterial
  - — Two-sided
- Texture Coordinate generation
- Fog computation

In Table 3-1, hardware accelerated operations are shown in bold face.

Table 3-1. Characteristics of Graphics Operations (Sheet 1 of 2)

Operation	Description
Whole Framebuffer Operations	
Clearing the buffers	**Back buffer and depth (Z) buffers cleared using the Intel740 chip**
Accumulation buffer	Accumulation buffer supported in software
Point Rasterization	
Single width points	**Rendered using Intel740 chip, using the line primitive**
Wide points	**Rendered using Intel740 chip, using the triangle primitive**
Anti-aliased points	Rendered in software
Line Rasterization	
Single width lines	**Rendered using Intel740 chip**
Wide lines	**Rendered using Intel740 chip**
Anti-aliased lines	Rendered in software
Stippled lines	Rendered in software
Triangle Rasterization	
Polygon Stippling	**Rendered using Intel740 chip**
Polygon Smoothing	ignored
Polygon culling	Hardware compatible culling to avoid cracks is done in software for maximum performance
Polygon mode	**The Intel740 driver determines the rendering of triangles as either FILL, LINE or POINT mode, and then uses the Intel740 chip hardware.**

Table 3-1. Characteristics of Graphics Operations (Sheet 2 of 2)

Operation	Description
Fragment Operations	
Fog	**Linearly interpolated fog factors are supported using Intel740 chip**
Textures	**Perspective-correct interpolation of texture coordinates using Intel740 chip**
Ownership test	Back buffers and depth (Z) buffers are exclusively owned by each application, so no ownership test is needed; however, the front buffer can be overwritten by other windows. Front buffer rendering is performed by the Intel740 chip whenever the front buffer is not occluded by other windows.
Scissor test	Optimized for rendering by the Intel740 chip whenever possible. Primitives that expand upon rasterization (i.e., wide lines) are not supported by hardware rendering when scissoring is ON.
Alpha test	**Supported by Intel740 chip**
Stencil test	Supported by the software renderer
Depthbuffer test	**Supported by Intel740 chip**
Alpha blending	**All blend modes are supported by Intel740 chip**
Dithering	**Supported by Intel740 chip**
Logical operations	Supported by the software renderer
Stencil operations	Supported by the software renderer
Buffer Write Controls	
RGB masking buffers	When (R=0, G=0, B=0) and (R=1, G=1, B=1), glColorMask is supported by the Intel740 chip. Other combinations are supported by the software renderer.
Depth buffer write mask	**glDepthMask is supported by Intel740 chip.**
DrawBuffer	**NONE: supported by Intel740 chip.** FRONT_AND_BACK: supported by software renderer. FRONT: front buffer rendering is done using the Intel740 chip when the front buffer is not occluded by other windows. **BACK: supported by Intel740 chip.**
Texture Mapping	
Image formats supported	A4, L6, L4A4, R5G6B5, R4G4B4A4.
Texture environment	Supports all OpenGL texture environment modes except GL_BLEND, which is supported only if the internal format of the texture image is GL_ALPHA.
GL_TEXTURE_MAG_FILTER	**Supported by Intel740 chip.**
GL_TEXTURE_MIN_FILTER	All filters except GL_LINEAR_MIPMAP_LINEAR and GL_NEAREST_MIPMAP_LINEAR are supported by Intel740 chip. **GL_LINEAR_MIPMAP_LINEAR and GL_NEAREST_MIPMAP_LINEAR are supported using texture dithering in the Intel740 chip.**
Texture border	Ignored

NOTE:
1. On Windows NT, The MCD allocates a separate front buffer per application.

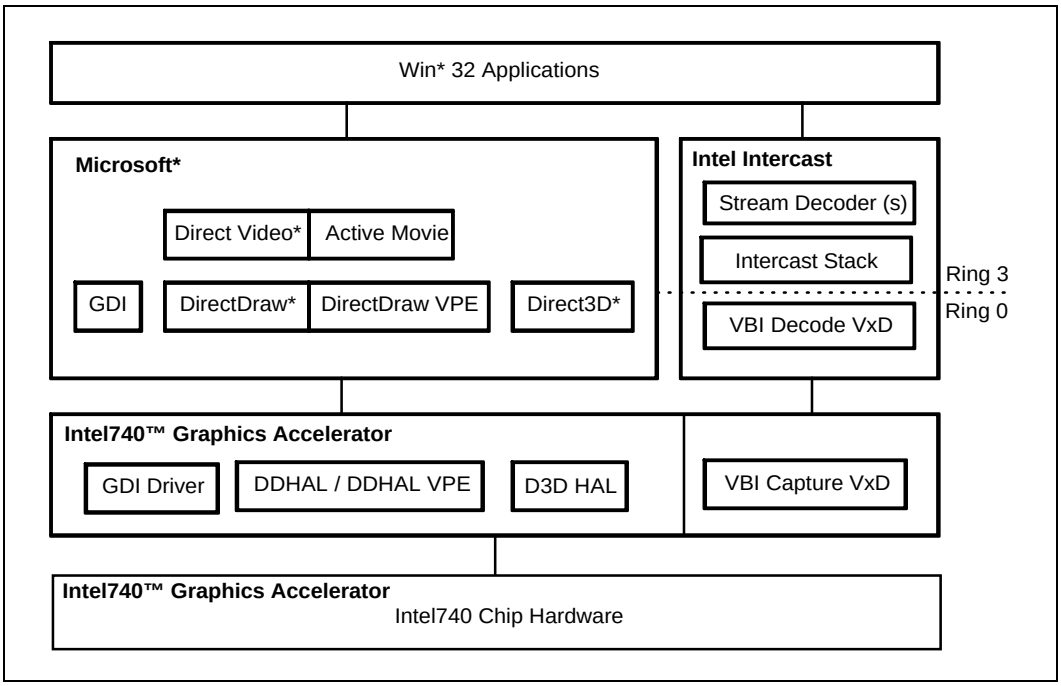
3.3 DirectX Programming Environment

This chapter explains the relationship between the Intel740™ graphics accelerator API and the Microsoft Windows* support driver environment (Microsoft Windows95*/Windows98*/ WindowsNT* 5.0). References are made to existing standards documents. Intel740™ graphics accelerator extensions or behaviors that differ from the standard are described in detail.

The Intel740™ graphics accelerator video support drivers include DirectDraw* (Overlay) driver, DirectDraw VPE driver, and VBI Capture VxD. The Intel740™ graphics accelerator DirectDraw Driver (DDHAL/DDHAL VPE) interfaces with the following external entities: Microsoft DirectX* API, and AGP Memory driver. The Intel740™ graphics accelerator VBI Capture VxD interfaces with the Intel VBI Decoder VxD, DDHAL VPE driver, AGP Memory driver. Table 3-3 shows the Intel740™ graphics accelerator driver architecture.

The Intel740™ graphics accelerator Direct3D device driver interfaces with the following external entities: Microsoft DirectX API, Intel740™ graphics accelerator 2D display driver, WIN32 GDI Escape Mechanism, Windows 95 Registry, and AGP Memory driver. The Configuration Applet along with any Diagnostic/Test applications will interface with the Intel740™ graphics accelerator Direct3D device driver through the GDI device-dependent graphics escapes defined by the driver. Figure 3-3 shows the Intel740™ graphics accelerator Direct3D driver architecture.

Figure 3-3. Intel740™ Graphics Accelerator Software Architecture



3.4 Windows Display Driver

3.4.1 Mini Display Driver

3.4.1.1 Structures Exported to GDI

Table 3-2. Device Technology—dpTechnology (Sheet 1 of 2)

Function	Supported
DT_PLOTTER(0)	
DT_RASDISPLAY(1)	✓
DT_RASPRINTER (2)	
Raster Capabilities—dpRaster	
RC_BITBLT (0001h)	✓ (8 BPP, 16 BPP, 24BPP)
RC_BANDING (0002h)	
RC_SCALING (0004h)	
RC_SAVEBITMAP (0040h)	
RC_PALETTE (0100h)	✓ (8 BPP)
RC_DIBTODEV (0200h)	✓ (8 BPP, 16 BPP, 24BPP)
RC_BIGFONT (0400h)	✓ (8 BPP, 16 BPP, 24BPP)
RC_STRETCHBLT (0800h)	✓ (8 BPP, 16 BPP, 24BPP)
RC_FLOODFILL (1000h)	
RC_STRETCHDIB (2000h)	✓ (8 BPP, 16 BPP, 24BPP)
RC_DEVBITS (8000h)	✓ (8 BPP, 16 BPP, 24BPP)
Level of text support the device driver provides—dpText	
TC_OP_CHARACTER (0001h)	
TC_OP_STROKE (0002h)	
TC_CP_STROKE (0004h)	✓
TC_CR_90 (0008h)	
TC_CR_ANY (0010h)	
TC_SF_X_YINDEP (0020h)	
TC_SA_DOUBLE (0040h)	
TC_SA_INTEGER (0080h)	
TC_SA_CONTIN (0100h)	
TC_EA_DOUBLE (0200h)	
TC_IA_ABLE (0400h)	
TC_UA_ABLE (0800h)	
TC_SO_ABLE (1000h)	
TC_RA_ABLE (2000h)	✓
TC_VA_ABLE (4000h)	

Table 3-2. Device Technology—dpTechnology (Sheet 2 of 2)

Function	Supported
Additional raster abilities—dpCaps1	
C1_TRANSPARENT (0001h)	
TC_TT_ABLE (0002h)	
C1_TT_CR_ANY (0004h)	
C1_EMF_COMPLIANT (0008h)	
C1_DIBENGINE (0010h)	✓
C1_GAMMA_RAMP (0020h)	✓
C1_ICM (0040h)	
C1_REINIT_ABLE (0080h)	
C1_GLYPH_INDEX (0100h)	✓
C1_BIT_PACKED (0200h)	
C1_BYTE_PACKED (0400h)	✓
C1_COLORCURSOR (0800h)	✓
C1_CMYK_ABLE (1000h)	
C1_SLOW_CARD (2000h)	
Polyline and line-drawing capabilities—dpLines	
LC_POLYGONSCANLINE (0001h)	✓
LC_POLYLINE (0002h)	✓
LC_WIDE (0010h)	
LC_STYLED (0020h)	✓
LC_WIDESTYLED (0040h)	
LC_INTERIORS (0080h)	
Polygon-, rectangle-, and scan-line drawing capabilities- dpPolygonals	
PC_ALTPOLYGON (0001h)	✓
PC_RECTANGLE (0002h)	
PC_WINDPOLYGON (0004h)	
PC_SCANLINE (0008h)	✓
PC_WIDE (0010h)	
PC_STYLED (0020h)	
PC_WIDESTYLED (0040h)	
PC_INTERIORS (0080h)	✓
PC_POLYPOLYGON (0100h)	
PC_PATHS (0200h)	

3.5 DirectDraw Display Driver Interface

This section explains the interfaces of Intel740™ graphics accelerator 2D drivers. It does not cover the whole 2D driver interface, since it is already defined by Microsoft* in the Windows95* or Windows98* DDK. This section specifies the interfaces of display driver, mini-VDD, DirectDraw HAL, DirectDraw VPE HAL and version information.

3.5.1 Directdraw Hal Capabilities

Table 3-3. dwCaps—Specifies Driver-Specific Capabilities

Function	Supported
DDCAPS_3D	✓
DDCAPS_ALIGNBOUNDARYDEST	
DDCAPS_ALIGNBOUNDARYSRC	
DDCAPS_ALIGNSIZEDEST	
DDCAPS_ALIGNSIZESRC	
DDCAPS_ALIGNSTRIDE	
DDCAPS_ALPHA	
DDCAPS_BANKSWITCHED	
DDCAPS_BLT	✓
DDCAPS_BLTCOLORFILL	✓
DDCAPS_BLTDEPTHFILL	✓
DDCAPS_BLTFOURCC	
DDCAPS_BLTQUEUE	
DDCAPS_BLTSTRETCH	
DDCAPS_CANBLTSYSMEM	✓
DDCAPS_CANCLIP	
DDCAPS_CANCLIPSTRETCHED	
DDCAPS_COLORKEY	✓
DDCAPS_COLORKEYHWASSIST	
DDCAPS_GDI	✓
DDCAPS_NOHARDWARE	
DDCAPS_OVERLAY	✓
DDCAPS_OVERLAYCANTCLIP	✓
DDCAPS_OVERLAYFOURCC	✓ (YUV4:2:2, RBG555 and RGB565)
DDCAPS_OVERLAYSTRETCH	✓
DDCAPS_PALETTE	
DDCAPS_PALETTEVSYNC	
DDCAPS_READSCANLINE	✓
DDCAPS_STEREOVIEW	
DDCAPS_VBI	
DDCAPS_ZBLTS	
DDCAPS_ZOVERLAYS	
DDCAPS_ZOVERLAYS	

Table 3-4. dwCaps2—Specifies More Driver-Specific Capabilities

Function	Supported
DDCAPS2_CERTIFIED	
DDCAPS2_NO2DDURING3DSCENE	
DDCAPS2_VIDEOPORT	✓
DDCAPS2_AUTOFLIPOVERLAY	✓
DDCAPS2_CANBOBINTERLEAVED	✓
DDCAPS2_WIDESURFACES	✓
DDCAPS2_NOPAGELOCKREQUIRED	

Table 3-5. dwCKeyCaps—Color Key Capabilities

Function	Supported
DDCKEYCAPS_DESTBLT	✓
DDCKEYCAPS_DESTBLTCLRSPACE	
DDCKEYCAPS_DESTBLTCLRSPACEYUV	
DDCKEYCAPS_DESTBLTYUV	
DDCKEYCAPS_DESTOVERLAY	✓
DDCKEYCAPS_DESTOVERLAYCLRSPACE	
DDCKEYCAPS_DESTOVERLAYCLRSPACEYUV	
DDCKEYCAPS_DESTOVERLAYONEACTIVE	✓
DDCKEYCAPS_DESTOVERLAYYUV	✓
DDCKEYCAPS_NOCOSTOVERLAY	✓
DDCKEYCAPS_SRCBLT	✓
DDCKEYCAPS_SRCBLTCLRSPACE	
DDCKEYCAPS_SRCBLTCLRSPACEYUV	
DDCKEYCAPS_SRCBLTYUV	
DDCKEYCAPS_SRCOVERLAY	
DDCKEYCAPS_SRCOVERLAYCLRSPACE	
DDCKEYCAPS_SRCOVERLAYCLRSPACEYUV	
DDCKEYCAPS_SRCOVERLAYONEACTIVE	
DDCKEYCAPS_SRCOVERLAYYUV	

Table 3-6. dwFXCaps—Specifies Driver-Specific Stretching and Effects Capabilities

Function	Supported
DDFXCAPS_BLTARITHSTRETCHY	
DDFXCAPS_BLTARITHSTRETCHYN	
DDFXCAPS_BLTMIRRORLEFTRIGHT	
DDFXCAPS_BLTMIRRORUPDOWN	
DDFXCAPS_BLTROTATION	
DDFXCAPS_BLTROTATION90	
DDFXCAPS_BLTSHRINKX	
DDFXCAPS_BLTSHRINKXN	
DDFXCAPS_BLTSHRINKY	
DDFXCAPS_BLTSHRINKYN	
DDFXCAPS_BLTSTRETCHX	✓
DDFXCAPS_BLTSTRETCHXN	
DDFXCAPS_BLTSTRETCHY	✓
DDFXCAPS_BLTSTRETCHYN	
DDFXCAPS_OVERLAYARITHSTRETCHY	✓
DDFXCAPS_OVERLAYARITHSTRETCHYN	
DDFXCAPS_OVERLAYMIRRORLEFTRIGHT	
DDFXCAPS_OVERLAYMIRRORUPDOWN	
DDFXCAPS_OVERLAYSHRINKX	
DDFXCAPS_OVERLAYSHRINKXN	
DDFXCAPS_OVERLAYSHRINKY	
DDFXCAPS_OVERLAYSHRINKYN	
DDFXCAPS_OVERLAYSTRETCHX	✓
DDFXCAPS_OVERLAYSTRETCHXN	
DDFXCAPS_OVERLAYSTRETCHY	✓
DDFXCAPS_OVERLAYSTRETCHYN	

Table 3-7. dwPalCaps—Specifies Palette Capabilities

Function	Supported
DDPCAPS_1BIT	✓
DDPCAPS_2BIT	✓
DDPCAPS_4BIT	✓
DDPCAPS_8BIT	✓
DDPCAPS_8BITENTRIES	
DDPCAPS_ALLOW256	
DDPCAPS_PRIMARYSURFACE	
DDPCAPS_PRIMARYSURFACELEFT	
DDPCAPS_VSYNC	

Table 3-8. ddsCaps.dwCaps—Specifies The Capabilities Of The Surface

Function	Supported
DDSCAPS_3D	✓ (Enabled if 3D is detected)
DDSCAPS_3DDEVICE	✓
DDSCAPS_ALLOCONLOAD	✓
DDSCAPS_ALPHA	
DDSCAPS_BACKBUFFER	✓
DDSCAPS_COMPLEX	✓
DDSCAPS_FLIP	✓
DDSCAPS_FRONTBUFFER	✓
DDSCAPS_HWCODEC	
DDSCAPS_LIVEVIDEO	✓
DDSCAPS_MIPMAP	✓
DDSCAPS_MODEX	✓
DDSCAPS_OFFSCREENPLAIN	✓
DDSCAPS_OVERLAY	✓
DDSCAPS_OWNDC	
DDSCAPS_PALETTE	✓
DDSCAPS_PRIMARYSURFACE	✓
DDSCAPS_PRIMARYSURFACELEFT	
DDSCAPS_SYSTEMMEMORY	✓
DDSCAPS_TEXTURE	✓
DDSCAPS_VIDEOMEMORY	✓
DDSCAPS_VISIBLE	✓
DDSCAPS_WRITEONLY	
DDSCAPS_ZBUFFER	✓
DDSCAPS_NONLOCALVIDMEM	✓

3.6 Direct3D Interface

3.6.1 Supported Direct3D Capabilities

Table 3-9. General Device Capabilities

Function	Supported
Device Color Model	
RGB	✓
Mono	✓
Device Capabilities	
FloatTLVertex	✓
SortIncreasingZ	
SortDecreasingZ	
SortExact	
ExecuteSystemMemory	
ExecuteVideoMemory	
TLVertexSystemMemory	
TLVertexVideoMemory	
TextureSystemMemory	
TextureVideoMemory	✓
Transform Capabilities	
Clip	
Lighting Capabilities	
RGBModel	
MonoModel	
Point	
Spot	
Directional	
ParallelPoint	
GLSpot	
Clipping	
True	
False	✓
Render Bit Depth	
16-bit	✓
Z Buffer Bit Depth	
16-bit	✓

Table 3-10. Texture Capabilities

Format	Width	Height	Bits Per Texel	R/Y Mask	G/U Mask	B/V Mask	Alpha Mask
RGB 565	1024	1024	16	F800h	07E0h	001Fh	0000h
RGBa 5551	1024	1024	16	7C00h	03E0h	001Fh	8000h
RGBa 4444	1024	1024	16	0F00h	00F0h	000Fh	F000h
YUV 422	1024	1024	8	F0h	0Ch	03h	00h
Palette Indexed 1	1024	1024	1				
Palette Indexed 2	1024	1024	2				
Palette Indexed 4	1024	1024	4				
Palette Indexed 8	1024	1024	8				

Table 3-11. Primitive Capabilities Supported (Sheet 1 of 3)

Capability	Lines	Triangles
Misc. Capabilities		
MaskPlanes		
MaskZ	✓	✓
LinePatternRep		
Conformant		
CullNone		✓
CullCW		✓
CullCCW		✓
Raster Capabilities		
Dither	✓	✓
Rop2		
Xor		
Pat		
Ztest	✓	✓
Subpixel	✓	✓
SubpixelX		
FogVertex	✓	✓
FogTable		
Stipple	✓	✓
Z/AlphaCompare Capabilities	**Z / Alpha**	**Z / Alpha**
Never	✓ / ✓	✓ / ✓
Less	✓ / ✓	✓ / ✓
Equal	✓ / ✓	✓ / ✓
LessEqual	✓ / ✓	✓ / ✓
Greater	✓ / ✓	✓ / ✓

Table 3-11. Primitive Capabilities Supported (Sheet 2 of 3)

Capability	Lines	Triangles
NotEqual	✓ / ✓	✓ / ✓
GreaterEqual	✓ / ✓	✓ / ✓
Always	✓ / ✓	✓ / ✓
Source/Destination Blend Capabilities	**Src / Dst**	**Src / Dst**
Zero	✓ / ✓	✓ / ✓
One	✓ / ✓	✓ / ✓
SrcColor	✓ / ✓	✓ / ✓
InvSrcColor	✓ / ✓	✓ / ✓
SrcAlpha	✓ / ✓	✓ / ✓
IncSrcAlpha	✓ / ✓	✓ / ✓
DestAlpha		
InvDestAlpha		
IncSrcAlpha	✓ / ✓	✓ / ✓
InvDestColor	✓ / ✓	✓ / ✓
SrcAlphaSat		
BothSrcAlpha	✓ / ✓	✓ / ✓
BothInvSrcAlpha	✓ / ✓	✓ / ✓
Shade Capabilities		
ColorFlatMono	✓	✓
ColorFlatRGB	✓	✓
ColorGouraudMono	✓	✓
ColorGouraudRGB	✓	✓
ColorPhongMono		
ColorPhongRGB		
SpecularFlatMono	✓	✓
SpecularFlatRGB	✓	✓
SpecularGouraudMono	✓	✓
SpecularGouraudRGB	✓	✓
SpecularPhongMono		
SpecularPhongRGB		
AlphaFlatBlend	✓	✓
AlphaFlatStippled	✓	✓
AlphaGouraudBlend	✓	✓
AlphaGouraudStippled		
AlphaPhongBlend		
AlphaPhongStippled		
FogFlat	✓	✓
FogGouraud	✓	✓

Table 3-11. Primitive Capabilities Supported (Sheet 3 of 3)

Capability	Lines	Triangles
FogPhong		
Texture Capabilities		
Perspective	✓	✓
Pow2	✓	✓
Alpha	✓	✓
Transparency	✓	✓
Border		
SquareOnly		
Texture Filter Capabilities		
Nearest	✓	✓
Linear	✓	✓
MipNearest	✓	✓
MipLinear	✓	✓
LinearMipNearest	✓	✓
LinearMipLinear	✓	✓
Texture Blend Capabilities		
Decal	✓	✓
Modulate	✓	✓
DecalAlpha	✓	✓
ModulateAlpha	✓	✓
DecalMask	✓	✓
ModulateMask		
Copy	✓	✓
Texture Address Capabilities		
Wrap	✓	✓
Mirror	✓	✓
Clamp	✓	✓

3.6.2 Supported RenderState

Table 3-12. DIRECT3D RenderState Hardware / Software Support (Sheet 1 of 3)

State	Supported in SW	Supported in HW	Values	Notes
ALPHAFUNC	✓	✓	NEVER LESS EQUAL LESSEQUAL GREATER NOTEQUAL GREATEREQUAL ALWAYS	
ALPHAREF	✓	✓	8-bit value	
ALPHATESTENABLE	✓	✓	TRUE / FALSE	
ANTIALIAS	✓	✓	SORTDEPENDENT / SORTINDEPENDENT	
ALPHABLENDENABLE	✓	✓	TRUE / FALSE	
CULLMODE	✓	✓	NONE CW CCW	
DESTBLEND	✓	✓	ZERO ONE SRCCOLOR INVSRCCOLOR SRCALPHA INVSRCALPHA DESTCOLOR INVDESTCOLOR BOTHSRCALPHA BOTHINVSRCALPHA	
DITHERENABLE	✓	✓	TRUE / FALSE	
FILLMODE	✓	✓	WIREFRAME - SOLID	
FOGENABLE	✓	✓	TRUE / FALSE	
FOGCOLOR	✓	✓	lower 24-bits of a 32-bit value	
FOGTABLEDENSITY	NO	NO		
FOGTABLEEND	NO	NO		
FOGTABLEMODE	NO	NO		
FOGTABLESTART	NO	NO		
LASTPIXEL	NO	NO	TRUE / FALSE	
LINEPATTERN	NO	NO	32-bit value	
MONOENABLE	✓	✓	TRUE / FALSE	
PLANEMASK	NO	NO	32-bit value	

Table 3-12. DIRECT3D RenderState Hardware / Software Support (Sheet 2 of 3)

State	Supported in SW	Supported in HW	Values	Notes
ROP2	NO	NO		
SHADEMODE	✓	✓	FLAT GOURAUD	
SPECULARENABLE	✓	✓	TRUE / FALSE	
SRCBLEND	✓	✓	ZERO ONE SRCCOLOR INVSRCCOLOR SRCALPHA INVSRCALPHA DESTCOLOR INVDESTCOLOR BOTHSRCALPHA BOTHINVSRCALPHA	
STIPPLEDALPHA	NO	NO		
STIPPLEENABLE	✓	✓	TRUE / FALSE	
STIPPLEPATTERN00-31	✓	✓	32-bit values	
SUBPIXEL	NO	NO		
SUBPIXELX	NO	NO		
TEXTUREADDRESS	✓	✓	WRAP MIRROR CLAMP	
TEXTUREHANDLE	✓	✓	32-bit value	
TEXTUREMAG	✓	✓	NEAREST LINEAR MIPNEAREST MIPLINEAR LINEARMIPNEAREST LINEARMIPLINEAR	
TEXTUREMAPBLEND	✓	✓	DECAL MODULATE DECALALPHA MODULATEALPHA DECALMASK COPY	
TEXTUREMIN	✓	✓	NEAREST LINEAR MIPNEAREST MIPLINEAR LINEARMIPNEAREST LINEARMIPLINEAR	
TEXTURE PERSPECTIVE	✓	✓	TRUE	

Table 3-12. DIRECT3D RenderState Hardware / Software Support (Sheet 3 of 3)

State	Supported in SW	Supported in HW	Values	Notes
WRAPUV	✓	✓	TRUE / FALSE	
WRAPV	✓	✓	TRUE / FALSE	
ZENABLE	✓	✓	TRUE / FALSE	
ZFUNC	✓	✓	NEVER LESS EQUAL LESSEQUAL GREATER NOTEQUAL GREATEREQUAL ALWAYS	
ZVISIBLE	NO	NO	TRUE / FALSE	
ZWRITEENABLE	✓	✓	TRUE / FALSE	

3.6.3 Supported RenderPrimitives

Table 3-13. DIRECT3D RenderPrimitive Hardware / Software Support

Primitive	Supported in SW	Supported in HW	Notes
POINT	✓	NO	Implemented as a 0 length line
LINE	✓	✓	
TRIANGLE	✓	✓	
SPAN	✓	NO	Implemented with a line
STRIP	✓	NO	Implemented with a triangle
FAN	✓	NO	Implemented with a triangle

3.7 Video Interface

All VfW Capture Messages are supported by the Intel740™ graphics accelerator video capture driver.

Table 3-14. VfW Capture Driver Capability

VfW Capture Message	Supported
DRV_LOAD	✓
DRV_FREE	✓
DRV_OPEN	✓
DRV_CLOSE	✓
DRV_ENABLE	✓
DRV_DISABLE	✓
DRV_QUERYCONFIGURE	✓
DRV_CONFIGURE	✓
DRV_INSTALL	✓
DRV_REMOVE	✓
DRV_GETVIDEOAPIVER	✓
DVM_GETERRORTEXT	✓
DVM_DIALOG	✓
DVM_PALETTE	✓
DVM_FORMAT	✓
DVM_PALETTERGB555	✓
DVM_SRC_RECT	✓
DVM_DST_RECT	✓
DVM_UPDATE	✓
DVM_CONFIGURE_STORAGE	✓
DVM_FRAME	✓
DVM_GET_CHANNEL_CAPS	✓
DVM_STREAM_INIT	✓
DVM_STREAM_FINI	✓
DVM_STREAM_GETERROR	✓
DVM_STREAM_GETPOSITION	✓
DVM_STREAM_ADDBUFFER	✓
DVM_STREAM_PREPAREHEADER	✓
DMV_STREAM_UNPREPAREHEADER	✓
DVM_STREAM_RESET	✓
DVM_STREAM_START	✓
DVM_STREAM_STOP	✓

3.8 GDI Escape Interface

The Intel740™ graphics accelerator Direct3D Driver supports the GDI Escape interface that allows dynamic alterations of operational parameters as well as debugging and performance monitoring. Access to these device capabilities which are specific to Intel740™ graphics accelerator 3D functionality is achieved using the following function call:

```
ExtEscape( HDC,  //handle to Windows device context
       int,    //Intel740™ graphics accelerator 3D escape function number (1234h)
       int,    //number of bytes in input structure
       LPCSTR, //pointer to input structure
               //typedef struct AubControlInBuffer
                   //   { DWord EscapeNumber;
                   //  DWordSubFunction;
                   //  DWordDataPointer;
                   //    }
       int,        //number of bytes in output structure
       LPSTR);     //pointer to output structure
                   //   typedef struct AubControlOutBuffer
                   //   { DWordEscapeNumber;
                   //  DWordSubFunction;
                   //  DWordDataPointer;
                   //    }
```

The following sections define the available subfunctions along with a definition for each DataPointer associated with the input and/or output structures. Data types which are in bold italic text are defined by Microsoft* in the DirectX documentation.

Table 3-15. Functionality Control

Sub-function	Description	AubControlInBuffer Data	AubControlOutBuffer Data
101h	Set State Variable	DWord StateNumber 01h-FFh - As defined by D3DRENDERSTATETYPE 100h - Texture LOD Bias 101h - Texture LOD Dither weight 102h - Alpha in Z buffer 103h - QWord fetch mode DWord StateValue	void
102h	Set Capabilities	D3DDEVICEDESC D3Dcapabilities	void
103h	Get Capabilities	void	D3DDEVICEDESC D3Dcapabilities
10Ah	Get AGP Config Registers	void	DWord Reg[3]

Table 3-16. Device Driver Debugging Control

Sub-function	Description	AubControlInBuffer Data	AubControlOutBuffer Data
200h	Set Debug Logging Level	DWord Level 0..MaxDebugLevel	void